OTP[1] Tabanlı DNA Şifreleme Yöntemi

Mir Mohammad Reza ALAVI MILANI[1] Hüseyin PEHLİVAN[2] Sahereh HOSEIN POUR[3]

[1,2,3]Karadeniz Üniversitesi, Bilgisayar Mühendisliği Bölümü, Trabzon

[1]e-posta: milani@ktu.edu.tr,[2] e-posta: pehlivan@ktu.edu.tr,[3] e-posta: hoseinpour@ktu.edu.tr

Özet: Günümüz bilgisayar destekli şifreleme teknikleri oldukça yüksek düzeyli bilgi gerektiren karmaşık güvenlik önlemleriyle yoğrulmuştur. Ancak, öncekilerden daha güvenli olduğu sanılan her bir yeni tekniğin zaman içerisinde başka güvenlik açıklarının bulunduğu ortaya çıkmaktadır. Genel olarak herhangi bir şifreleme yönteminin kırılamaz olmadığını, sonlu bir süre sonunda şifresinin çözülebileceğini söyleyebiliriz. Şifreleme güvenliğini artırmada DNA kavramlarını kullanan yöntemlerin sayısı hızla artmaktadır; bazı şifreleme yöntemleri DNA kavramlarının biyolojik özelliklerinden, bazıları da modellemesinden yararlanırlar. Bu çalışmada, lojistik haritanın rastgele özelliklerinden yararlanılarak oluşturulan bir DNA_OTP dizisi yardımıyla bir DNA şifreleme yöntemi geliştirilmiştir. Lojistik haritadan oluşan sayıların rasgele özellikli ve anahtardan üretilen başlangıç değerine duyarlı olması önerilen algoritmanın güvenliğini yükseltir.

Anahtar: Şifreleme, Lojistik Harita, Rasgele , DNA , OTP , DNA_OTP .

A OTP Based DNA Encryption Method

Abstract: Today's computer-aided encryption techniques are equipped with very complicated and complex security measures requiring high-level knowledge. But, with every coming days, we are witnessing other security problems in respect of each new technique that has been thought to be more secure than the previous ones. In general, we can say that that any encryption method cannot become "unbreakable" and any password can break in a limited period of time. The number of encryption methods which use DNA concepts are increasing gradually; some encryption methods are based on the biological features of DNA concepts, the others benefit from their modeling. In this study, we propose a DNA encryption algorithm, using the Henon chaotic systems and a DNA_OTP list created from the logistic properties of random maps. Due to the random properties of the logistic map and the sensitive nature of the primary value obtained from the key, the proposed algorithm is very secure.

Keywords: Encryption, logistic map, random numbers, DNA, OTP , DNA_OTP.

1. Giriş

Son yıllarda, kaos fizik, matematik, mühendislik, biyoloji, kimya ve ekonomi gibi birçok bilim dalında büyük bir ilgi uyandırmaya başlamıştır. Örneğin, son on yıldan beri ayrık kaotik dinamik sistemlerinde şifreleme gereksinimleri için yaygınca kullanılmaktadır [1-3].

Ayrıca internet ve kablosuz ağlar üzerinde şifreleme ve güvenli iletim sistemlerinin önemi giderek artmakla birlikte, AES, DES, IDEA, RSA [4] gibi klasik algoritmaların kullanılması uygun görünmemektedir. Bu zamana kadar, çok sayıda dijital kaotik şifrelemesinde ileri sürülmüştür [5-8]. Genel bir tasarım ilkesi olarak, şifrelemede temel blokların düzeltilmesi doğrusal olmayan fonksiyonlarla yapılmaktadır [9]. Ayrık ve sürekli zaman kaotik sistemlerini birleştiren daha karmaşık bir sistem Guan ve arkadaşları tarafından tasarlanmıştır [10]. Başka bir yöntem şifreleme hız ve güvenliğini artırmak için geliştirilmiştir [11].

Literatürde kaotik sistem özelliklerini kullanan algoritmalar da bulunmaktadır [12,13]. Kaotik algoritmalar değişik bir yol kullanırlar; bu algoritmalar, kaotik sistemlerin başlangıç değeri, sistem parametreleri ve rastgelelik özelliklerine dayalı olarak çok basittir ve hesaplama maliyetleri azdır. Bu nedenle kaotik algoritmalarla yapılan sistemlerin hızı daha yüksek olabilmektedir.

Diğer taraftan, DNA şifreleme, kriptografi araştırmalarında yeni ve çok umut verici bir yön olmuş. Gehani et al. [14] DNA iplikleri ile One-Time Pad (OTP) kriptografi kullanarak bir görüntü şifreleme algoritması sunmuşlardır. DNA olağanüstü bilgi yoğunluğuna sahiptir ve büyük bir OTP saklamak için oldukça uygundur. Bu yöntem OTP depolama sorununu çözmede etkili olabilir. Kang Ning [15] ise iyi şifreleme ve gerçek biyolojik deneyler yoluyla değil, bir sözde DNA

[1] One Time Pad

şifreleme yöntemi önermiştir. Son olarak, kaotik sistemleri kullanan metotlarda, sistemin anahtar değişikliğine çok hassas olduğu göz önüne alındığında, bu sistemlerin daha güvenli olduğunu söyleyebiliriz.

Bu çalışma aşağıdaki gibi yapılandırılmıştır. 2. Bölümde, lojistik haritanın özellikleri tartışılmıştır. 3. Bölüm DNA hesaplamaya ve 4. Bölüm OTP şifrelemeye ayrılmıştır. 5. Bölümde yeni bir şifreleme yöntemi önerilirken 6. Bölümde önerilen yöntem bir örnek üzerinden açıklanmıştır.

2. Lojistik Harita

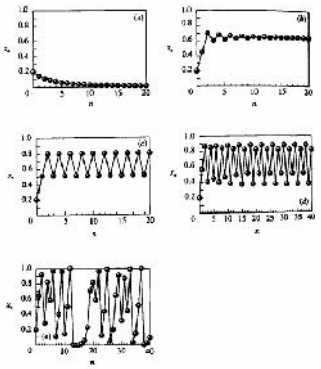
Bilgi taşımak için kaotik sinyallerin kullanılması, ilk olarak Hayes ve arkadaşları tarafından 1993 yılında ortaya atılmıştır [16]. Kaos tabanlı şifreleme programları temelde kaotik denklemleri kullanarak sözde rastgele sayı üreteçleri gibi uzun bir rastgele sayı dizisi üretip bu dizi ile bir düz görüntüyü şifrelerler [17]. Basit ve en çok çalışılan doğrusal olmayan sistemlerden biri lojistik haritadır. Bu sistem aslında 1838 yılında Pierre Franois Verhulst tarafından demografik bir model olarak tanıtılmıştır. 1947 yılında, Ulam ve von Neumann [18] rastsal sayı üreteci olarak lojistik haritayı çalıştı. Görüntülerin şifrelenmesinde, lojistik haritalar, onların başlangıç koşullarına hassas bağımlılığı, rastgeleye benzer davranış göstermesi ve tekrarlı olmayan özellikleri içermesinden dolayı S-box kutularının yerine kullanılır [17]. Kaos tabanlı şifreleme programları temelde, kaotik haritaları kullanarak rastsal sayı üreteçleri olarak bir uzun rastgele sayı dizisi üreterek düz görüntüyü bu rastgele sayılarla şifrelerler [17,19].

Lojistik harita aşağıdaki gibi verilir:

$$X_{n+1} = \lambda X_n (1 - X_n) \qquad (1)$$

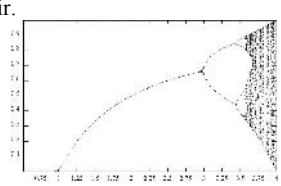
Burada sırasıyla $X_n \in (0,1)$ ve λ sistem değişkeni ve parametresi, n ise yineleme sayısıdır. Böylece, bir başlangıç değeri x0 ve bir parametre k alarak, $\{X_n\}_{n=0}^{\infty}$ serisi hesaplanır. Bu çalışmada, X_0 ve λ değerleri lojistik haritanın başlangıç değerleri olarak adlandırılacaktır. Bu başlangıç değerlerinin, özellikle λ değerinin, lojistik haritada çok önemli bir işlevi vardır. Bu önemi göstermek için aşağıdaki durumu ele alalım:

Şekil 1'de $X_0 = 0.2$ ve farklı λ değerleri için lojistik haritanın zaman içerisinden yinelemelere bağlı olarak değişimi gösterilmektedir.



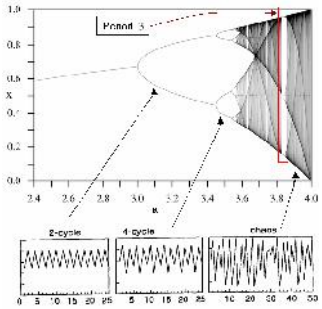
Şekil 1: (a) λ=0.9, (b) λ=2.6, (c) λ=3.2, (d) $\lambda = 3.57$, (e) λ=4

Şekil 1'e göre lojistik harita $\lambda = 0.9, 2.6, 3.2$ değerlerinde değil, $\lambda = 3.57, 4$ değerlerinde kaotik özellikler gösterir. Lojistik haritanın farklı λ değerlerinden ne kadar etkilendiği çatallanma[1] diyagramı ile Şekil 2'de gösterilmiştir. Bu bir λ fonksiyonu olarak, lojistik haritanın bir komplosudur. $0 \le \lambda \le 1$ için elde edilen çözüm sadece bir sabit noktadır. $1 < \lambda \le 3$ için, yine sabit bir nokta vardır. $3 < \lambda \le 3.75$ arasında, haritanın iki katına çıkarılması sergilenir. $3.75 < \lambda < 4$ için, harita kaotik olur. Nihayet, $\lambda = 4$ durumunda, kaos 0-1 arasında değişen çeşitli değerlerden oluşabilmektedir.



Şekil 2: Çatallanma (bifurcation) diyagramı

Şekil 3'de, Şekil 2'de açıkça görünmeyen 2.4 ile 4.0 noktaları arasındaki harita özellikleri daha ayrıntılı biçimde gösterilmiştir.



Şekil 3: Şekil 2 nin $2.4 \le \lambda \le 4.0$ diyagramı

Bu çalışmada, rastgele sayıların oluşturulması için lojistik haritayı aşağıdaki gibi kullanacağız:

$$X_{n+1} = \lambda X_n (1 - X_n),\ X_n \in (0,1),\ \lambda \in (3.9996, 4] \qquad (2)$$

3. DNA Hesaplama ilkeleri

Moleküler hesaplama olarak da bilinen DNA (Deoksiriboz Nükleik Asit) hesaplama, Adlemanın açan çalışmaya dayalı kitlesel paralel hesaplamada yeni bir yaklaşımdır.

DNA molekülü, dört nükleik asit bazlarından, yani A (adenine), T (thymine), G (guanine) ve C (cytosine)'den oluşur. Bu nükleikler Watson-Crick kurallarına göre sadece A ile T ve C ile G çiftleri birleşerek bir araya gelirler.

Bir santimetre küpe 10 trilyondan fazla DNA molekülü sığar. Bu hacimde DNA ile, 10 terabayt (1000 gigabayt) bilgi kaydedebilir ve bir anda 10 trilyon hesaplama yapabiliriz. Bu nedenle DNA özellikleri çeşitli yöntemlerde kullanım yeri bulmuştur. Bazı yöntemler paralel hesaplama özelliklerinden yararlanırken, diğerleri bir bellek olarak DNA'yı kullanmıştır.

Ayrıca hesaplama teorisinde de DNA'dan yararlanabiliriz. Örneğin, yeni bir sistem oluşturabiliriz. Bu sistemin alfabesini $\Sigma = \{A, C, G, T\}$ varsayalım. Ayrıca bu sistemin alfabesi aşağıdaki özelliklere sahiptir:

$$\overline{A} = T, \overline{C} = G, \overline{G} = C, \overline{T} = A \qquad (3)$$

Bu sistemde tüm n uzunluğuna sahip olan diziler tanımlaması aşağıdaki denklem (4) gibi tanımlanabilir:

$$Z = \Sigma^n = \{< a_1, a_2, a_3, ..., a_n > | \, a_i \in \Sigma, i = 1,2,..n\} \qquad (4)$$

Bu yeni sistemde çeşitli işlemler yapılabilir; diğer sistemdeki verileri bu sisteme çevirip, burada işlem yapmak mümkündür. Bu sistemde yapılan işlemlerin sonuçları da tekrar önceki sisteme geri dönüşebilir. Ancak herhangi bir sistemden bu sisteme dönüşü ve tersi, homomorfizma fonksiyonlarla mümkündür. Örneğin bir ASCII sistemden bu sisteme çevirme işlemi şöyle gerçekleşebilir:
İlk önce ASCII veriler ikili şekilde yazılır (her karakter, 8 bite), ve daha sonra elde edilen veriler aşağıdaki homomorfizma fonksiyon ile DNA sistemine dönüşür.

h(00) = A, h(01) = C, h(10) = G, h(11) = T (5)

Görüldüğü gibi bu fonksiyon özel değildir ve değişik şekillerde yapılabilir. Ancak denklem (3)'yi her zaman sağlamalıdır. Böylece ASCII sisteminde veya hatta görüntüde her pikselin değerlerini DNA sistemine dönüştürebiliriz.
Bilindiği gibi DNA sistemindeki dizilerin uzunluğu, ASCII sistemindeki dizinin dört katıdır. Yani:

$| Z_{DNA} | = 4 * | Z_{Ascii} |$ (Z_{DNA} DNA sistemindeki dizinin uzunluğu , Z_{Ascii} ASCII sistemindeki dizinin uzunluğu).

Örneğin "Example" dizisi önce ikili sisteme (01000101, 01111000, 01100001, 01101101, 01110000, 01101100, 01100101) ve daha sonra DNA sistemine (CACC, CTGA, CGAC, CGTC, CTAA, CGTA, CGCC) biçiminde dönüşür.

4. Tek Kullanımlık Karakter Dizisi (One-time Pad)

Bu basit şifreleme yönteminde rastgele üretilen bir karakter (harf veya rakam) dizisi kullanılarak şifreleme yapılır. Düz metin (plain text) içinde yer alan her karakter, üretilen dizide karşısına düşen karakterle işleme sokularak (örneğin modüler toplama işlemi) şifreli mesaj elde edilir. Mesajı çözmek için rastgele dizinin bilinmesi gereklidir. Bu yönteme Vernam şifreleme yöntemi de denir. Örneğin:

Düz Metin : KRIPTOLOJINET
Rastgele Dizi : DEFRYPLCNMLJK
Şifreli Mesaj : KUSOPZPNMDGOK

Bu yöntemin güvenliği rastgele üretilen diziye bağlıdır. Bu dizi gerçekten rastgele üretilmelidir, eğer bir kurala bağlı olarak üretilirse ve bu kural saldırgan tarafından bilinirse sistem kırılabilir. Bu tehdit dışında sistem mükemmel bir şifreleme sistemidir ve ilk olarak 1917'de Vernam tarafından tasarlanmış ve "teletype" makinelerinde kullanılmıştır.
Bu yöntemde düz metnin bit sayısı kadar uzunluğunda tamamen rastgele bir anahtar dizisi ile düz metin bitlerinin d-ya'sına (dışarlayıcı ya, exor) dayanır. Düz metin P, anahtar dizisi K, düz metin bit sayısı N ise şifreli metin olan C'nin bitleri şu şekilde belirlenir: $C_i = P_i \oplus K_i$, i=1,...,N.

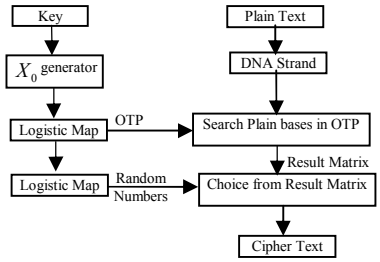
Bu sistem mükemmel gizliliği sağlar, yani Sadece Şifreli Metin saldırısı uygulamak sonsuz hesapsal güç sahibi olunsa dahi imkansızdır. Yalnız, mükemmel gizliliği sağlamak için anahtar dizisinin sadece bir kere kullanılması şarttır.
Bu çalışmada, kaotik sistemlerden yararlanarak, bir küçük boyutlu bir anahtar kelimesinden bir büyük anahtar dizisi üreteceğiz ve bu dizini OTP olarak adlandıracağız.

5. Önerilen Yöntem

Bu çalışmada önerilen yöntemde, ilk önce düz metin 3. Bölümde bahsedildiği gibi bir DNA dizisine dönüştürülür, sonra bu dönüştürülmüş düz metinden dörderli bazlar alınarak bir OTP dizisinde aranır ve bulunduğu yerin indeksi bir matris dizisinin, düz metinde olan karakterin eşdeğer sütununa yerleştirilir. Daha sonra bu matrisin sırayla her sütunundan rasgele bir sayı seçilir ve şifrelenmiş veri olarak bir diziye eklenir.
Bu işlemleri yaparken aşağıda verilen konuların dikkate alınması gerekir:

- OTP dizisi, anahtar kelime ve kaotik harita kullanılarak oluşturulmaktadır.
- OTP dizisinin boyutunu n varsayalım; böylece arama işlemi aşağıdaki gibi diziden bir sıra içinde yapılır ve bulunduğu yer 'i' olarak kaydedilir: $\{< a_i, a_{i+1}, a_{i+2}, a_{i+3} > | \, a_i \in OTP, i = 1,2,..., n-3\}$
- Son kısımda kullanılan rastgele sayılar da kaotik haritadan elde edilecektir.

Önerilen yöntemin akış şeması Şekil 4'te gösterilmektedir:



Şekil 4 – Önerilen yöntemin akış şeması

Şekil 4'te göründüğü gibi, ilk önce anahtar kelimeden bir x_0 üretilir ve daha sonra bu x_0 değeri, lojistik haritanın başlangıç değeri olarak rastgele sayı üreten bölüme aktarılır. Lojistik haritanın kullanımında başlangıç değeri olarak X_0, algoritmanın anahtarından seçilir. Anahtar kelime, en fazla n*8 bitten oluşan bir kelime veya herhangi bir veri olabilmektedir. Bu veriyi n ASCII karakteri olarak (her biri 8 bit) $K_0, K_1, K_2, ..., K_n$ biçiminde ifade edebiliriz ve buradaki her bir K_i'yı da 8 bit'ten meydana geldiğinden $K_{i1}, K_{i2}, K_{i3}, ..., K_{i8}$ gibi gösterebiliriz. x_0 değerinin 0 ve 1 aralığında olma şartını göz önüne alarak denklem (5)'deki gibi basit bir komut yazabiliriz:

$$X_0 \leftarrow [K_{11} * 2^{8n-1} + K_{12} * 2^{8n-2} + K_{13} * 2^{8n-3} + ... + K_{n8} * 2^0 +] / 2^{8n} \qquad (6)$$

Böylece x_0, 0 ve 1 aralığında bir değer olarak hesaplanır.
Aşağıdaki C kodu ile bir anahtar kelimeden (KeyStr) bir X_0 değeri hesaplanmaktadır.

```
double Createx0(String KeyStr){
  int n,k=8;
  double sum=0;
  n=KeyStr.Length();
  for(int i=1;i<=n;i++,k+=8)
     sum+=(double)KeyStr.operator [](i)*pow(2,k);
  sum+=(double) KeyStr.operator [](1)*pow(2,k);
  k+=8;
  return  sum/pow(2,k);
}
```

Şekil 5 – Anahtar kelimesinden başlangıç değeri üreten C++ kodu

Kaotik özellikleri kullanan şifreleme yöntemleri, genellikle kaotik sistemlerden oluşturulan rastgele sayılar kullanırlar [10,12]. Bu çalışmada önerilen yöntemde lojistik harita denklem (2)'deki gibi kullanılarak 0 ve 1 aralığında rastgele sayılar üretilmiştir.

Yöntemi işlemlerde kullanırken λ'nın değeri 3.99999 olarak seçilmiştir.

Önerilen yöntemin iki farklı kısmında bu rastgele sayılara ihtiyaç duyulur:

5.1 OTP dizisini oluşturmak,

5.2 Sonuç matrisinin her sütunundan birini seçerken belli aralıkta rastgele sayılar üretmek.

OTP dizisini oluşturmak için, elde edilen X_n'ler [0,1] arasında olacağından, bu aralığı 0,3 aralığına taşımak için, X_n'leri 4 ile çarparız ve tam kısmını kullanırız. Böylece {0,1,2,3} sayı kümesinden bir elemanı X_n değeri yerine hesaplayabilir ve bu kümenin izdüşümü olan {A,C,G,T}'yi DNA dizisine ekleyebiliriz. OTP dizisini oluştururken aşağıdaki konuları göz önüne almalıyız:

- OTP dizisinin boyutu çok büyük olmalı,
- ASCII kodlarının dörtlü DNA izdüşümleri bu dizide aynı sayıda tekrarlanmalı.

Son olarak, sonuç matrisinden seçerken, rastgele sayıları her sütunda yer alan verilerin sayısı kadar üretilmelidir.

6. Deneysel Sonuçlar

Bu çalışmada deneysel olarak küçük boyutlu bir düz metnin şifrelemesi yapılmıştır. Bunun için OTP dizisinin boyutunu 32767 varsayarak, lojistik haritanın başlangıç değeri x_0 ile denklem (2)'yi 32767 kere kullanarak 32767 elemanlı $\{OTP_0, OTP_1, OTP_2, ..., OTP_{32767}\}$ kümesi oluşturulmuştur. Bu kümenin oluşturulmasında x_0, Şekil 5'deki Createx0 fonksiyonu ile üretilmiştir. Ayrıca λ'nın değeri burada 3.999999 seçilmiştir. Bu kümeyi oluştururken, lojistik haritadan elde edilen x_i'ları 256 ile çarptıktan sonra, onun tam kısmını elde ederiz. Böylece elde edilen sayılar 0 ve 255 aralığında bulunur. Ancak bu verileri OTP kümesine eklerken 0'dan 255'e kadarki sayıların sayısının eşit miktarda olmasına dikkat edilmelidir. Böylece tüm değerlerden OTP dizisinden eşit sayıda bulunabilir.

Daha sonra tüm düz metni (burada örnek olarak "Example") bir ikili diziye, daha sonra (5)'teki homomorfizma fonksiyonu ile bir DNA dizisine dönüştürürüz.

"Example" → { 69 , 120 , 97 , 109 , 112 , 108 , 101 } →
"01000101011110000110000101101101011100000110110001100101"
→ "CACCCTGACGACCGTCCTAACGTACGCC" .

Bu diziden ilk dört baz alarak, OTP dizisinde ararız ve bulunduğu yerlerin indekslerini ilk karakterin aday verileri olarak sonuç matrise ekleriz. Bu işlemi düz metnin diğer dörder bazlarına da uygularız. Böylece her karaktere karşı elde edilen aday sayılar belirlenir.

Örneğin :

69 → {23 , 602 , 1865 , 4443 , 4480 , 7400 , 11006 , 13254 , 14015 , ... , 29746 , 30868 , 31502 , 31800 , 31901 , 32401}

120 → {258 , 789 , 927 , 1295 , 2954 , 3045 , 3098 , 3181 , 3207 , 3361 , 3763 , … , 30087 , 30097 , 30110 , 30438}

97 → {102 , 609 , 1009 , 1421 , 2308 , 4012 , 4126 , 4219 , 4412 , 4910 , ... , 30472 , 31090 , 31487 , 32304 , 32426}

109 → {86 , 196 , 358 , 592 , 1496 , 4103 , 4292 , 4871 , 5207 , 5991 , ... , 29145 , 29937 , 31520 , 31902 , 32416}

112 → {289 , 409 , 479 , 893 , 1763 , 2037 , 4362 , 4906 , 5131 , 5874 , ... , 28097 , 29957 , 30570 , 31159 , 32604}

108 → {107 , 188 , 284 , 957 , 2109 , 2509 , 3142 , 3691 , 4217 , ... , 30457 , 31207 , 31761 , 32017 , 32418 , 32569}

101 → {607 , 815 , 1026 , 1592 , 1937 , 2719 , 3131 , 3592 , 4296 , ... , 27695 , 29333 , 29739 , 31512 , 31910 , 32101}

Görüldüğü gibi her karaktere karşı 128 indeks olmalıdır. Bundan dolayı, OTP'de kullanan lojistik harita (denklem (2)) devam ettirir ve her karaktere karşı bir rastgele sayı, 0 ve 128 aralığında üretir ve aday dizilerinden birini, o karakterin şifresi olarak seçeriz.

Örneğin:

69 → 70. indeks → 16215
120 → 112. indeks → 26073
97 → 35. indeks → 10271
109 → 6. indeks → 4103
112 → 97. indeks → 25016
108 → 74. indeks → 15184
101 → 3. indeks → 1026

Böylece "Example" düz metin olarak {16215 , 26073 ,10271 , 4103 , 25016 , 15184 , 1026} şifreli metne dönüştürülür.

7. Sonuç

Bu çalışmada, şifrelemesi yapılacak verileri bulunduğu sistemden bir DNA sistemine taşıyarak ve yeni sistem içinde kaotik haritaların rastgele özelliklerini kullanarak şifreleyen bir yöntem önerilmiştir. Çalışmanın diğer bir önemli konusu, tamamen bir rastgele OTP dizisi üretmektir. Üretilen OTP dizisinin, lojistik haritanın başlangıç değerine ve anahtar kelimeye bağlılık düzeyi açıkça görülmektedir. Anahtar kelimedeki küçük bir değişiklikle, çok farklı bir OTP üretilmesi söz konusudur. Ayrıca üretilen OTP bir DNA dizisi olduğundan, bu veriler DNA molekülünün yüksek taşıma kapasitesine sahip olma özelliklerinden faydalanarak, alıcı tarafına çok kolayca transfer edilebilir. Son olarak, önerilen

yöntemde OTP dizisinin kullanımıyla şifreleme güvenliğinin yükseltileceği sonucunu çıkarmak mümkündür.

8. Kaynakça

[1] R. Brown, L.O. Chua, International Journal of Bifurcation and Chaos 6 (1996) 219.

[2] J. Fridrich, International Journal of Bifurcation and Chaos 8 (1998) 1259.

[3] L.M. Pecora, T.L. Carroll, Physical Review Letters 64 (1990) 821.

[4] Daemen J, Sand B, Rijmen V. The design of Rijndael: AES – the advanced encryption standard. Berlin: Springer-Verlag; 2002.

[5] S. Behnia, A. Akhshani, S. Ahadpour, H. Mahmodi, A. Akhavan, Physics Letters A 366 (2007) 391.

[6] S. Behnia, A. Akhshani, H. Mahmodi, A. Akhavan, Chaos Solitons & Fractals 35 (2008) 408.

[7] M.S. Baptista, Physics Letters A 240 (1998) 50.

[8] S. Behnia, A. Akhshani, H. Mahmodi, A. Akhavan, International Journal of Bifurcation and Chaos 18 (2008) 251.

[9] Preneel B. Design principles for dedicated hash functions. In: Fast software encryption, Cambridge security workshop, Lecture notes in computer science, vol. 809, Springer, Berlin; 1993. p. 71–82.

[10] Guan Z H, Huang F, Guan W. Chaos based image encryption algorithm. Phys Lett A 2005;346:153–7.

[11] Menezes AJ, van Oorschot PC, Vanstone SA. Handbook of applied cryptography. CRC Press; 1997.

[12] Pareek NK, Patidar V, Sud KK. Image encryption using chaotic logistic map. Image Vision Comput 2006;24:926–34.

[13] Chen G, Mao Y, Chui CK. A symmetric image encryption scheme based on 3D chaotic cat maps. Chaos Solitons Fract 2004;21:749–61.

[14] A. Gehani, T.H. LaBean, J.H. Reif, DNA-based cryptography. DIMACS series in discrete mathematics, Theoretical Computer Science 54 (2000) 233_249.

[15] Kang Ning, A pseudo DNA cryptography method, arXiv:0903.2693.

[16] Hayes S, Grebogi C, Ott E. Communicating with chaos. Phys Rev Lett 1993;70(20):3031–4.

[17] Pisarchik AN, Flores-Carmona NJ, Carpio-Valadez M. Encryption and decryption of images with chaotic map lattices. Chaos: Interdiscipl J Nonlinear Sci2006;16(3):033118.

[18] Ulam SM, von Neumann J. On combination of stochastic and deterministic processes. Bull Am Math Soc 1947;53:1120.

[19] Fridrich J. Symmetric ciphers based on two-dimensional chaotic maps. Int J Bifurcat Chaos 1998;8:1259–84.